On the unsupervised analysis of domain-specific Chinese texts

On the unsupervised analysis of domain-specific Chinese texts

Ke Deng[a], Peter K. Bol[b], Kate J. Li[c], and Jun S. Liu[a,d,1]

[a]Center for Statistical Science & Department of Industry Engineering, Tsinghua University, Beijing 100084, China; [b]Department of East Asian Languages & Civilizations, Harvard University, Cambridge, MA 02138; [c]Sawyer Business School, Suffolk University, Boston, MA 02108; and [d]Department of Statistics, Harvard University, Cambridge, MA 02138

Edited by Wing Hung Wong, Stanford University, Stanford, CA, and approved March 25, 2016 (received for review August 21, 2015)

With the growing availability of digitized text data both publicly and privately, there is a great need for effective computational tools to automatically extract information from texts. Because the Chinese language differs most significantly from alphabet-based languages in not specifying word boundaries, most existing Chinese text-mining methods require a prespecified vocabulary and/or a large relevant training corpus, which may not be available in some applications. We introduce an unsupervised method, top-down word discovery and segmentation (TopWORDS), for simultaneously discovering and segmenting words and phrases from large volumes of unstructured Chinese texts, and propose ways to order discovered words and conduct higher-level context analyses. TopWORDS is particularly useful for mining online and domain-specific texts where the underlying vocabulary is unknown or the texts of interest differ significantly from available training corpora. When outputs from TopWORDS are fed into context analysis tools such as topic modeling, word embedding, and association pattern finding, the results are as good as or better than that from using outputs of a supervised segmentation method.

word discovery | text segmentations | EM algorithm | Chinese history | blogs

Due to the explosive growth of the Internet technology and the public adoption of the Internet as a main culture media, a large amount of text data is available. It is more and more attractive for many researchers to extract information from diverse text data to create new knowledge. Biomedical researchers can gain understanding on how diseases, symptoms, and other features are spatially, temporally, and ethnically distributed and associated with each other by mining research articles and electronic medical records. Marketers can learn what consumers say about their products and services by analyzing online reviews and comments. Social scientists can discover hot events from news articles, web pages, blogs, and tweets and infer driving forces behind them. Historians can extract information about historical figures from historical documents: who they were, what they did, and what social relationships they had with other historical figures.

For alphabet-based languages such as English, many successful learning methods have been proposed (see ref. 1 for a review). For character-based languages such as Chinese and other East Asian languages, effective learning algorithms are still limited. Chinese has a much larger "alphabet" and vocabulary than English: *Zhonghua Zihai Dictionary* (2) lists 87,019 distinct Chinese characters, of which 3,000 are commonly used; and the vocabulary of Chinese is an open set when named entities are included. Additionally, morphological variations in Latin-derived languages (e.g., uppercase or lowercase letters, tense and voice changes), which provide useful hints for text mining, do not exist in Chinese. Because there is no space between Chinese characters in each sentence, significant ambiguities are present in deciphering its meaning.

There are two critical challenges in processing Chinese texts: (*i*) word segmentation, which is to segment a sequence of Chinese characters into a sequence of meaningful Chinese words and phrases; and (*ii*) word and phrase discovery, a problem similar to named entity recognition in English whose goal is to identify unknown/unregistered Chinese words, phrases, and named entities from the texts of interest. In practice, word segmentation is often entangled with word discovery, which further compounds the difficulty. Many available methods for processing Chinese texts focus on word segmentation and often assume that either a comprehensive dictionary or a large training corpus (usually texts manually segmented and labeled from news articles) is available. These methods can be classified into three categories: (*i*) methods based on word matching (3), (*ii*) methods based on grammatical rules (4–6), and (*iii*) methods based on statistical models [e.g., hidden Markov model (7) and its extensions (8), maximum entropy Markov model (9), conditional random field (10–12), and information compression (13)]. These methods, especially the ones based on statistical models, work quite well when the given dictionary and training corpus are sufficient and effective. However, once the target texts are considerably different from the training corpora or the actual vocabulary has a significant portion outside the given dictionary, such as those historical documents accumulated throughout ancient China that contain many unregistered technical words and use some different grammatical rules, performances of these supervised methods drop dramatically.

To our best effort, we have only found limited literature on unsupervised Chinese word discovery and segmentation (14–18), and none has discussed context analyses based on unsupervised segmentation results. Some methods designed for speech recognition (19–22) are related to this problem but cannot be directly applied for processing Chinese texts. Some of the aforementioned supervised methods can discover new words, but it happens only when the discovered words have very similar patterns to words in the training corpus. We here propose an unsupervised method, top-down word discovery and segmentation (TopWORDS), to simultaneously segment any given Chinese texts and discover words/phrases without using a given dictionary or training corpus. Our method is based on a statistical model termed the "word dictionary model" (WDM), which has arisen from the text-mining community (14, 23–26). Although the WDM is not new, effective and scalable methods for analyzing Chinese texts based on it have not been known, which is likely due to two key challenges: the initiation of the unknown dictionary and the final selection of the inferred words.

Different from previous methods, which typically infer the final dictionary by growing from a small initial dictionary containing word candidates of one or two characters long, TopWORDs starts with a large, overcomplete, initial dictionary and prunes it down to a proper size based on statistical estimation principles. Previous methods also did not have the final word selection step, of which the consequence

Significance

We propose top-down word discovery and segmentation (TopWORDS), an unsupervised tool for Chinese word (and phrase) discovery, word ranking, and text segmentation. We show that pipelines formed by combining TopWORDS with context analysis tools can help researchers quickly gain insights into new types of texts without training and recover almost all interesting features of the text that a well-trained supervised method can find.

Author contributions: K.D. and J.S.L. designed research; K.D., P.K.B., K.J.L., and J.S.L. performed research; K.D. and J.S.L. analyzed data; and K.D., K.J.L., and J.S.L. wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission.

[1]To whom correspondence should be addressed. Email: jliu@stat.harvard.edu.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1516510113/-/DCSupplemental.

is to include too many false or partial words. TopWORDS uses a statistical model selection strategy to score each inferred word, giving rise to a natural ranking and the final selection of the words. Fig. 1 illustrates the general architecture of TopWORDS. We show in *Results* how analysis pipelines that combine TopWORDS with a content analysis method such as topic modeling, word embedding, and association mining, can help us quickly gain insights into new domain-specific Chinese texts without training.

Theory and Methods

WDM. A sentence is a sequence of basic characters of a language, but is read and understood via higher-order units, i.e., words, phrases, idioms, and regular expressions, which in our context are all broadly defined as "words." Let $\mathcal{A}=\{a_1,\cdots,a_p\}$ be the set of basic "characters" of the language of interest. In English, it is the alphabet containing only 26 letters, whereas in Chinese it is the set of all distinct characters appearing in the text, often of the size of thousands. A word w is defined as a sequence of elements in $\mathcal{A}$, i.e., $w=a_{i_1}a_{i_2}\cdots a_{i_J}$. Let $\mathcal{D}=\{w_1,w_2,\cdots,w_N\}$ be the vocabulary (dictionary) for the texts of interest. WDM regards each sentence S (and the whole text) as a concatenation of words drawn randomly from $\mathcal{D}$ with sampling probability θ_i for word w_i. With $\boldsymbol{\theta}=(\theta_1,\cdots,\theta_N)$ representing the word use probability vector, where $\sum_{i=1}^{N}\theta_i=1$, the probability of generating a K-word (segmented) sentence $S=w_{i_1}w_{i2}\cdots w_{i_K}$ from WDM is as follows:

$$P(S|\mathcal{D},\boldsymbol{\theta})=\prod_{k=1}^{K}\theta_{i_k}. \quad [1]$$

This model can be traced back to ref. 23, and was used in ref. 14 to do Chinese word segmentation and in ref. 27 to analyze genomic sequences. Compared with the complexity and subtleties of natural languages, WDM is clearly a rough approximation. Although ignoring long-range dependencies among words and phrases in texts, WDM provides a computationally feasible statistical framework for unsupervised text analysis.

Word Segmentation Based on WDM. In English texts, words are recognizable due to the employment of spacing between adjacent words, whereas in Chinese no spacing is used within a sentence. For unsegmented Chinese text T, we let $\mathcal{C}_T$ denote the set of all segmented sentences corresponding to T permissible under dictionary $\mathcal{D}$. Then, under model [1], we have the following:

$$P(T|\mathcal{D},\boldsymbol{\theta})=\sum_{S\in\mathcal{C}_T}P(S|\mathcal{D},\boldsymbol{\theta}); \quad [2]$$

and the conditional probability,

$$P(S|T;\mathcal{D},\boldsymbol{\theta})\propto P(S|\mathcal{D},\boldsymbol{\theta})\mathbf{1}_{S\in\mathcal{C}_T}, \quad [3]$$

which measures how likely T can be segmented into S under WDM. The maximum-likelihood (ML) segmentation of T is thus defined as follows:

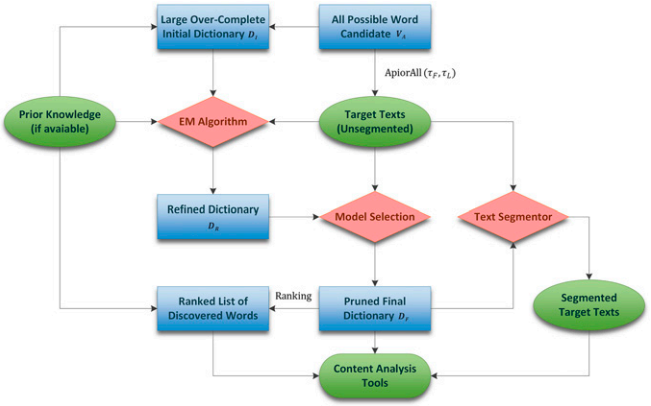

Fig. 1. Flowchart of the TopWORDS algorithm: collections of words are shown in blue rectangles, algorithms are in red diamonds, and text-related materials and tools are highlighted in green. The "Prior Knowledge" and "Target Texts" are the founding nodes, and outputs are the "Ranked List" of words and the "Segmented Texts," which are then fed into a Content Analysis tool to gain contextual insights.

$$S^*=\arg\max_{S\in\mathcal{C}_T}P(S|T;\mathcal{D},\boldsymbol{\theta}).$$

A more robust approach than the ML segmentation is to average over all possible segmentations of T. To explain, we let $I_k(S)=1$ if the segmentation S puts a word boundary behind the kth basic character of T, and let $I_k(S)=0$ otherwise. Then, the score

$$\gamma_k(T)=\sum_{S\in\mathcal{C}_T}P(S|T;\mathcal{D},\boldsymbol{\theta})\cdot I_k(S) \quad [4]$$

measures the total probability of having a word boundary behind position k of T considering all possible ways of segmenting T. A segmentation of T can be created by placing a word boundary behind the kth character of T if $\gamma_k(T)$ is greater than a given threshold τ_γ. We refer to this strategy as the posterior expectation (PE) segmentation. Note that a PE segmentation may contain components that are not proper words in $\mathcal{D}$, although this rarely happens in practice if τ_γ is not too small (e.g., $\tau_\gamma>0.5$). Hence, we use PE segmentation unless it contains improper words, in which case we use ML segmentation.

TopWORDS. In unsupervised text analyses, it is a main challenge to discover the unknown dictionary $\mathcal{D}$ from a given set of unsegmented texts $\mathcal{T}=\{T_1,\cdots,T_n\}$. The first effort to tackle the problem dates back to Olivier's "word grammar" (23), a stepwise method that starts with an initial dictionary with only single-character words and iterates between estimating word use frequencies $\boldsymbol{\theta}$ for a given dictionary $\mathcal{D}$ and adding new words to the current dictionary. The algorithm is terminated when no new words can be found. However, due to the lack of a principled method and computational resources at that time, both steps are ad hoc approximations with suboptimal statistical properties. Later on, computer scientists and linguists improved Olivier's method and proposed a few information-phrasesbased methods (24–26). The approach was further improved in refs. 14 and 27 by using the maximum-likelihood estimation (MLE) procedure and applied to genomics and Chinese text analysis. The WDM was also generalized to a more complicated Markov dictionary model in ref. 28. All of these methods discover new words based on a "bottom-up" heuristics, which recursively adds to the current dictionary $\mathcal{D}$ new candidates made up from concatenations of existing words.

Although the bottom-up approach is successful for English texts and genomic sequence analyses, it is too expensive for Chinese texts because both dictionary $\mathcal{D}$ and alphabet $\mathcal{A}$ are very large. TopWORDS employs a "top-down" strategy for word discovery. It starts with a large, overcomplete, dictionary $\mathcal{D}$ consisting of all strings whose length is no greater than τ_L and frequency in the texts of interest no smaller than τ_F (τ_L and τ_F are user-specified thresholds). This step is achieved by the ApriorAll algorithm in ref. 29. All basic characters in $\mathcal{A}$ are put into $\mathcal{D}$ as well. Assigning each word w_i a use frequency parameter θ_i, TopWORDS uses the EM algorithm (30) to obtain the MLE of $\boldsymbol{\theta}=(\theta_1,\theta_2,\ldots,\theta_N)$. The main difficulty in estimating $\boldsymbol{\theta}$ lies in the ambiguity of text segmentation. The E-step of the EM algorithm needs to sum over all possible segmentations, which fortunately can be achieved by using a dynamic programming scheme with a time complexity of $O(Len(T)\cdot\tau_L)$ (*SI Appendix, Technical Details*).

A good choice of the starting value of $\boldsymbol{\theta}$ for the EM algorithm is the normalized observed counts vector. Because the initial $\mathcal{D}$ contains many nonwords and composite words, many estimated θ's are zero or very close to zero. We thus can trim down $\mathcal{D}$ to a much-smaller–sized dictionary $\mathcal{D}^*$. In fact, in each EM iteration, TopWORDS prunes away candidate words whose estimated use frequencies are close enough to zero (e.g., $<10^{-8}$). This strategy can greatly speed up the EM algorithm with little impact on the quality of the final results.

It is easy to integrate prior knowledge into TopWORDS as follows: (*i*) if a string corresponds to a known word a priori, we automatically put it in dictionary $\mathcal{D}$, overriding other criteria used by the algorithm; (*ii*) if a string is known to be an improper word, we remove it from the initial $\mathcal{D}$; and (*iii*) if a properly segmented training corpus is available, its contribution to the count of each candidate word w_i is directly combined with the contribution from unsegmented texts in the E-step of the EM algorithm.

Ranking and Selecting the Discovered Words. Word candidates that survive at the end of the EM algorithm can be further ranked. Let $\hat{\boldsymbol{\theta}}$ be the MLE obtained by TopWORDS based on unsegmented texts: $\mathcal{T}=\{T_1,\ldots,T_n\}$. For each $w_i\in\mathcal{D}$, we define $\hat{\boldsymbol{\theta}}_{[w_i=0]}\triangleq(\hat{\theta}_1,\cdots,\hat{\theta}_{i-1},0,\hat{\theta}_{i+1},\cdots,\hat{\theta}_N)$, and compute w_i's significance score ψ_i as the logarithm of the likelihood ratio statistics between the model $(\mathcal{D},\hat{\boldsymbol{\theta}})$ and model $(\mathcal{D},\hat{\boldsymbol{\theta}}_{[w_i=0]})$:

$$\psi_i=\sum_{j=1}^{n}\log\frac{P(T_j|\mathcal{D},\hat{\boldsymbol{\theta}})}{P(T_j|\mathcal{D},\hat{\boldsymbol{\theta}}_{[w_i=0]})}. \quad [5]$$

A large ψ_i means that w_i is statistically important for WDM to fit the target texts $\mathcal{T}$. Asymptotically, $2\psi_i$ follows the χ^2_1 distribution if θ_i is indeed 0, based

on which word candidates that are not statistically significant can be identified and removed. This strategy is computationally efficient and works well in practice. Word selection can also be achieved by maximizing the penalized likelihood function below with a much higher computational cost:

$$f(\boldsymbol{\theta}) = \sum_{j=1}^{n} \log P(T_j|\mathcal{D}, \boldsymbol{\theta}) + h(\boldsymbol{\theta}), \quad [6]$$

where the regularization term $h(\boldsymbol{\theta})$ penalizes a dictionary by its size, and different $h(\boldsymbol{\theta})$ lead to different model selection criterion, such as Akaike information criterion (AIC) (31), Bayesian information criterion (BIC) (32), least absolute shrinkage and selection operator (LASSO) (33), etc.

A natural choice to rank the discovered words is by their significance scores. In some text analysis tasks, however, the comparison and contrast of K different target texts are of interest. To highlight the specific content of each target text, we can also use the estimated "relative frequency,"

$$\phi_{ik} = \hat{\theta}_{ik} \Big/ \sum_{j=1}^{K} \hat{\theta}_{ij}, \quad [7]$$

as a ranking criterion, i.e., how a word w_i is enriched in one target text compared with the background "average" text. Here, $\hat{\theta}_{ik}$ is the estimated use frequency of word w_i from the kth target text. Our discovered words in the next section are ranked by either significance scores or relative frequencies.

Analysis Pipelines. The data-driven vocabulary discovered by TopWORDS and its resulting segmented texts can be used as inputs to other text mining tools for Chinese where a given dictionary and/or segmented texts are needed. In the topic modeling pipeline, outputs of TopWORDS are fed to a topic modeling algorithm such as latent-Dirichlet allocation (LDA) (34–36). Topic models can be viewed as a generalization of the WDM and is also a "bag-of-words"–type model. It assumes that each "bag" (i.e., an article) is composed of several latent "topics" with each topic represented by a probability vector on all words. As shown in *Results*, in an analysis of blog posts written by eight Chinese bloggers, this pipeline can provide us with topics that accurately reflect the themes of each blogger.

Association rule mining (37) and its extensions (38–41) seek to discover statistically significant patterns of co-occurrence of multiple items (words in our case) within a domain, such as a sentence or a paragraph. In the association-mining pipeline, TopWORDS is combined with these tools to capture higher-level information from texts. As shown in our analysis of *History of the Song Dynasty*, this pipeline can provide us useful information regarding social connections of important political figures of the Song dynasty in China's history.

Word embedding (42) is a method recently developed for neural network learning of alphabet-based text data. Its idea is to represent each word by a Euclidean vector of 50∼200 dimension learned during the training process of the neural network. It is recently shown in ref. 43 that the embedding algorithm in ref. 42 is equivalent to the singular value decomposition of a word co-occurrence matrix and can be resolved in a computationally efficient way with no supervision. This embedding implicitly encodes contextual information regarding the word use and can also capture some information that association mining tools do. Because Chinese words are ambiguous in the text, this method has not been popularly used for Chinese text analysis. We demonstrate in *Results* that our word-embedding pipeline naturally combines TopWORDS with the word-embedding algorithm of ref. 43 to learn word types, structures, and themes of the target Chinese texts.

TopWORDS can also be extended to fit more complicated scenarios. For example, Markov dependence can be introduced to certain words in the dictionary if a strong prior knowledge is available. In addition, if texts of interest are obtained from different sources that have different use preferences of certain known words, we can extend the current model by allowing source-specific parameters for these special words. Fig. 1 displays how TopWORDS organizes its each step.

Results

Discovering English Words and Phrases in *Moby Dick*. As a proof of concept, we applied TopWORDS to the English novel *Moby Dick* ($\sim 2\times 10^5$ word tokens) in the same fashion as Bussemaker et al. (27), who analyzed only the first 10 chapters of the novel (135 chapters in total). We first converted the novel to a long string of lowercase letters containing no spaces, numbers, or punctuation marks (string size $\approx 10^6$). Starting from an initial overcomplete dictionary with $\sim 3\times 10^5$ word candidates ($\tau_L = 13$ and $\tau_F = 3$), TopWORDS took about 10 min to converge to the final dictionary (all computations in this article were done on a Dell PowerEdge 1950 computer with 2.83-GHz CPU and 8-GB RAM). It contains ∼11,000 words, among which 6,349 are authentic English words, 3,438 are concatenations of English words (e.g., "mobydick," "atlast"), and 1,610 are fragments of words (e.g., "ing," "tion"). Moreover, about 75% word boundaries of the original texts are correctly recovered (sensitivity, 75%; specificity, 87%). Considering that a majority of the missed word boundaries are those within English phrases such as "mobydick" and "atlast," the "adjusted sensitivity," which ignores the missed word boundaries within proper English phrases, is greater than 85%.

The novel has ∼17,000 distinct English words, among which only 6,730 words have appeared more than twice. Because rare words (i.e., words that appear no more than twice) are not "discoverable" in an unsupervised way by default, it is of interest to see how the result can be improved if we treat these rare words as prior knowledge. We thus augmented the initial dictionary $\mathcal{D}$ with all rare words and let them evolve with other word candidates generated via enumeration. In this case, only 108 discovered words were fragments, and only 842 (fewer than 5%) true words were missed (most of which are words such as "moby" and "dick" that tend to be recognized as one composed word "mobydick"). The sensitivity, adjusted sensitivity, and specificity of word segmentation increased to 76%, 95%, and 99%, respectively, which is comparable to the current best supervised methods (8–13). More details can be found in *SI Appendix*, Table S1, Fig. S1, and Data File A.

We also applied TopWORDS to synthesized texts simulated based on the WDM with the true dictionary consisting of the 1,000 most frequently used words in *Moby Dick*. When the size of the simulated text is larger than 200,000 English letters, TopWORDS can discover the underlying word dictionary and segment the simulated texts almost perfectly. The average sensitivity and specificity of word discovery across 100 independent runs were 98% and 93%, respectively, and the average error rate for text segmentation was less than 2% (sensitivity, 98.6%; specificity, 99.7%).

Unsupervised Versus Supervised. 《红楼梦》[*The Story of the Stone* (SoS), also known as *The Dream of the Red Chamber*], written by Cao Xueqin about 250 y ago, is indisputably the most outstanding Chinese classical novel. It contains ∼1 million Chinese characters, being composed of ∼4,500 distinct ones. It manages an extraordinarily large number of fictional characters: more than 700 people's names were created, of which 371 had appeared more than twice and are referred to as "frequent names." Starting from an initial dictionary with ∼78,000 word candidates ($\tau_L = 8$ and $\tau_F = 3$), TopWORDS took about 100 s to converge to the final dictionary containing ∼17,000 nontrivial words, and segmented the novel with ∼290,000 word boundaries. More than 90% of the discovered words are semantically meaningful.

We applied the word-embedding algorithm of ref. 43 to embed the top 2,000 words discovered and ranked by TopWORDS in the space spanned by the top 200 eigenvectors of the word co-occurrence matrix. The distance (or cosine of the angle) between a pair of words in this embedding space represents their "similarity," both semantically and contextually. Fig. 2*A* shows the multidimensional scaling (MDS) plot (44) of the top 100 discovered words. MDS seeks to place a set of high-dimensional points in a plane for visualization so that between-point distances are maximally preserved. It is striking to see that all important characters' names (colored in red) among the top 100 words fall naturally into one cluster without any prior training, indicating that accurate recognition of types of words discovered by TopWORDS is possible. The detailed protocol of the word-embedding pipeline is provided in the *SI Appendix*, Table S2.

We also applied two popular supervised tools for processing Chinese texts, the Stanford Parser (SP) developed by the Stanford

Natural Language Processing Group (nlp.stanford.edu/software/lex-parser.shtml) and the Language Technology Platform (LTP) developed by researchers from Harbin Institute of Technology (www.ltp-cloud.com), to the full texts of SoS. SP is trained with the Penn Chinese Treebank (www.cis.upenn.edu/~chinese/) and LTP with the PKU corpus (www.icl.pku.edu.cn). SP and LTP yielded quite different results for SoS, although both methods claimed high precision in their own tests with their chosen training and testing corpora. For SoS, SP and LTP reported ~43,000 and ~38,500 words, respectively, of which only ~23,000 are in common. They also predicted ~370,000 and ~405,000 word boundaries, respectively, of which ~337,000 are in common. These differences imply that the text of SoS is quite different from the training corpora used by SP and LTP, and also suggest that text segmentation and word discovery for domain-specific Chinese texts are still challenging.

Among the ~290,000 word boundaries predicted by TopWORDS, ~240,000 (82%) were also predicted by SP, and ~250,000 (86%) by LTP, indicating that TopWORDS had a good specificity in text segmentation. TopWORDS performed better than SP and LTP in identifying important technical terms, although it predicted fewer words and word boundaries due to its tendency to preserve long phrases. To illustrate, we focus on the discovery of the 371 frequent names in SoS. TopWORDS successfully captured 345 names with only 26 missing (7%), significantly outperforming SP and LTP, which missed 59 (16%) and 89 (24%) frequent names, respectively. SP and LTP failed to identify more frequent names because some of the names in SoS are quite different from ordinary Chinese names in their training corpora. For example, many servants in the novel are named with phrases in classic poems and often do not have family names. In contrast, TopWORDS worked robustly and was adaptive to characteristics of SoS. More detailed results and comparisons are provided in *SI Appendix*, Table S3 and Data File B.

Analyzing a Traditional Chinese Book. Traditional Chinese evolves over time, differs from modern Chinese in many ways, and cannot be easily analyzed using available text-mining tools. As a demonstration, we applied TopWORDS to the full text of《宋史》(*History of the Song Dynasty*, abbreviated as HSD),

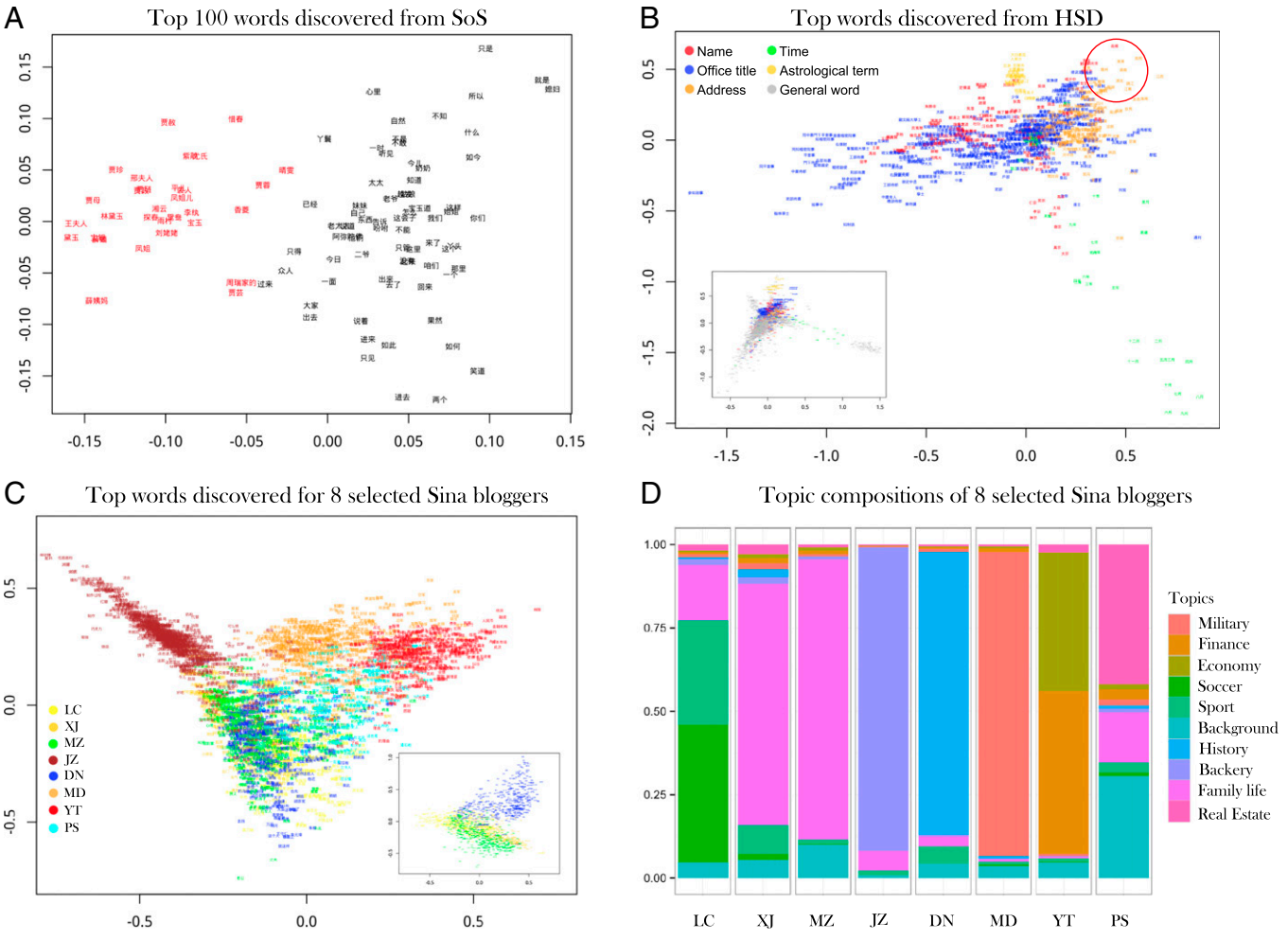

Fig. 2. Results from analysis pipelines of TopWORDS. (*A*) We selected the top 2,000 words discovered and ranked by TopWORDS in SoS and projected them to a 200-dimensional space using a word-embedding algorithm (43). Plotted here are the multidimensional scaling (MDS) (44) plot of the top 100 words after embedding. Names of characters in SoS were marked red. (*B*) Top 5,000 words found in HSD by TopWORDS were subject to the word-embedding analysis, and the MDS of all of the technical terms within the top 2,000 words were plotted, while all of the top 2,000 words were plotted in the *Inset*. The color codes are shown on the top left corner of the figure. (*C*) From the analysis results of Sina.com blogs, We chose top 1,000 words from each blogger to form a set of 4,500 distinct words, which are then mapped to a 200-dimensional space by word embedding. Shown here is the MDS plot for the ~2,500 blogger-specific words, and the *Inset* shows the MDS for the four bloggers that are not well separated in the main plot. (*D*) All blogs were pooled and analyzed by TopWORDS and the outputs were fed into the LDA algorithm (34) with $K = 10$ topics. Each colored bar represents the topics composition of one blogger, and the topics are color coded with their main themes indicated on the side bar.

which contains 496 chapters and records the history of China from 960 A.D. to 1279 A.D. It has about 5.3 million Chinese characters composed of ~10,000 distinct ones. A large number of technical words (e.g., names, addresses, office titles, time labels) that are not found in most authoritative Chinese dictionaries appear in this book. TopWORDS started the initial dictionary $\mathcal{D}$ with ~220,000 word candidates ($\tau_L = 10$, $\tau_F = 5$) and ended with a dictionary of more than 73,000 words within 25 min. Over 90% of the discovered words are correct (according to our random survey), which can be a name, an office title, an address, a time label, or other common words. Some interesting top words are the following: 簽書樞密院事 (office title), 執政進拜加官 (promotion), 如太白 (an astrological term), 庚戌 (a time label), and 蔡京 (a minister's name).

We applied word embedding to embed the top 5,000 words TopWORDS reported in a 200-dimensional space, and made MDS plots for the top 2000 of them. The larger plot in Fig. 2*B* shows all technical words ranked within the top 2000, and the *Inset* shows all of the top 2,000 words. Each color represents a specific type of words, such as people's names (red), office titles (blue), etc. The figure shows some general clustering trend for words of the same type (color). When more points (words) are processed by MDS and shown (as in the *Inset*), some outlying points drive the visualization and compress other points. We can observe more delicate structures when we zoom in with fewer points shown (as in the main figure). Besides word classification, the word-embedding pipeline provides us more insights into the content. For example, if a person's name is close to a geographic address, it tends to imply that the person had spent much of his life there. People whose names are tightly clustered together were often close colleagues.

Outputs of TopWORDS can also be fed to a relationship-finding method, such as association rule mining (37–40) and the theme dictionary model (TDM) (41). These methods tend to find more refined and tighter associations than those revealed by word embedding. By using only the discovered technical terms from the segmented texts of each paragraph of HSD, we constructed ~50,000 "baskets" of technical terms, one basket corresponding to one paragraph of HSD. Applying TDM to these baskets, we obtained more than 1,000 association patterns of technical terms, of which 90% are well supported by history. Similar to the implications of those clusters in Fig. 2*B* except being more certain and specific, an association between names often corresponds to colleagues or enemies; an association between a name and an office title indicates that the person was appointed at that position during his professional career; and an association between a name and an address suggests that the person either was born or once worked there. For example, the association pattern {程頤, 程顥, 張載, 周敦頤} (four names) is supported by the fact that they were key contemporaries establishing an influential branch of Confucianism. The patterns {張俊, 岳飛, 劉光世} and {張俊, 岳飛, 韓世忠} correspond to the names of four famous generals of the Southern Song period who were key to the revival of the country. These association patterns are also supported by the word-embedding result as highlighted by the cluster of names in the red circle in Fig. 2*B*. Interestingly, the addresses in that circle happen to be places where these generals fought most of their battles. More detailed results are provided in *SI Appendix*, Table S4 and Data File C.

Mining Online Chinese Texts from Sina Blogs. As one of the most popular portals and a major blog service provider in China, Sina.com has attracted many bloggers and readers. Popular bloggers are followed and commented by millions of fans on a daily basis, providing a valuable source for studying modern China. However, unknown vocabulary and flexible grammar of online texts create challenges to the application of available text-mining methods.

We selected eight famous Sina bloggers from the club of "top 100 Sina bloggers" ranked by the number of page views. They have diverse backgrounds and writing interests, which make them a representative sample of the most influential bloggers in China. These bloggers are as follows: 李承鵬 (Li Chengpeng or LC, journalist and soccer commentator; page views, 3.2 billion), 徐靜蕾 (Xu Jinglei or XJ, movie star and director; page views, 3.1 billion), 木子李 (Mu Zili or MZ, romantic fiction writer; page views, 3.0 billion), 君之 (Jun Zhi or JZ, pastry chef and entrepreneur; page views, 2.6 billion), 当年明月 (Dang Nian Ming Yue or DN, writer and historian; page views, 2.3 billion), 马鼎盛 (Ma Dingsheng or MD, journalist and military commentator; page views, 1.6 billion), 叶檀 (Ye Tan or YT, economist; page views, 1.5 billion), and 潘石屹 (Pan Shiyi or PS, president of one of the largest real estate companies in China; page views, 0.7 billion).

Numbers of blog posts produced by the eight bloggers vary from a few hundred to 2,000. We combined all blog posts written by each blogger as one corpus, and all corpora combined have 13 million characters including punctuation marks. We applied TopWORDS to these eight corpora separately with $\tau_L = 8$ and $\tau_F = 3$. About 170,000 words were discovered, of which only 30% are contained in the largest online Chinese dictionary Sogou Dictionary (搜狗词库; pinyin.sogou.com/dict/). A distinct feature of TopWORDS is to preserve idioms and regular expressions. For example, the word "参加" ("join") is often associated with different activities. TopWORDS reports not only the original word "join," which is listed in regular dictionaries, but also common phrases such as "join today," "join formally," "join an activity," which are not covered by a dictionary. One can argue that such phrases are not "words" at the linguistic level, but they reflect more on the content and can potentially be more useful for downstream studies. Full lists of top words discovered for each blogger and segmented corpora are in *SI Appendix*, Table S5 and Data File D.

We chose the top 1,000 words from each blogger's corpora to form a union set with 4,500 distinct words, among which ~2,500 are blogger specific. Word embedding was applied to the union set. Fig. 2*C* shows the MDS plot of the 2,500 blogger-specific words. We can see that words used by PS (cyan), YT (red), JZ (brown), and MD (orange) form their own clusters in the main figure. The *Inset* displays the MDS for the other four bloggers' words only. Words of DN (blue), LC (light yellow), and MZ (green) also form their own clusters, but words of XJ (dark yellow) appear to be buried in the middle, suggesting that XJ's writing style is perhaps "middle-of-the-road." This is consistent with our topic modeling results below. The result also suggests an article recommendation system using outputs from TopWORDS.

We next experimented with the topic modeling pipeline by pooling all segmented blog posts of the eight bloggers into one corpus and applying the LDA (34) to it. Each segmented blog post was treated as a "document" and the discovered words as "words." We tested the number of topics K ranging from 5, 10, 15, to 70 and found that the results were consistent, although more detailed substructures could be captured when K was larger. Fig. 2*D* illustrates the topic compositions of each blogger. MD's topic is predominantly military related, whereas JZ writes almost all about cooking and DN mostly about history. MZ and XJ have a lot of similarities in their topics (mainly female and relationship issues related), which is also supported by the word-embedding results discussed earlier. The other three bloggers typically write about two main topics. More details about the discovered topics are provided in *SI Appendix*, Table S5.

Conclusion

We propose an unsupervised method, TopWORDS, that can achieve word discovery and text segmentation simultaneously for domain-specific Chinese texts. The method can also be readily applied to English and other alphabet-based languages for discovering regular expressions, idioms, and special names by treating each English word as a character, which can be more informative for downstream analyses. Analysis pipelines that

combine TopWORDS with another high-level context analysis method, such as word embedding, topic modeling, and association rule mining, can reveal key characteristics of the texts of interest. Compared with existing methods for mining Chinese texts, the TopWORDS pipeline exhibits the following advantages: (*i*) it works stably for domain-specific Chinese texts, for which neither training data nor a proper dictionary is available; (*ii*) it is powerful in discovering unknown or unregistered words, especially long phrases; (*iii*) it is based on a probabilistic model, which facilitates rigorous statistical inferences with efficient computation; (*iv*) it incorporates prior information easily (when available) for better performance; and (*v*) it can generate useful frontline features (for all languages) and extract key characteristics for text understanding.

ACKNOWLEDGMENTS. This work is partially supported by National Science Foundation Grant DMS-1208771 and National Natural Science Foundation of China Grant 11401338.

1. Cambria E, White B (2014) Jumping NLP curves: A review of natural language processing research. *IEEE Comput Intell Mag* 9(2):48–57.
2. (1994) *Zhonghua Zihai Dictionary* (Zhonghua Book Company, Beijing).
3. Chen KJ, Liu SH (1992) Word identification for Mandarin Chinese sentences. *Proceedings of the Fourteenth International Conference on Computational Linguistics* (Association for Computational Linguistics, Stroudsburg, PA), Vol 1, pp 101–107.
4. Geutner P (1996) Introducing linguistic constraints into statistical language modeling. *Proceedings of the Fourth International Conference on Spoken Language* (Institute of Electrical and Electronics Engineers, Piscataway, NJ), Vol 1, pp 402–405.
5. Chen A (2003) Chinese word segmentation using minimal linguistic knowledge. *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing* (Association for Computational Linguistics, Stroudsburg, PA), Vol 17, pp 148–151.
6. Shu X (2014) Words segmentation in Chinese language processing. PhD thesis (University of Illinois at Urbana–Champaign, Urbana, IL).
7. Sproat R, Gales W, Shih C, Chang N (1996) A stochastic finite-state word-segmentation algorithm for Chinese. *Comput Linguist* 2(3):377–404.
8. Zhang HP, Yu HK, Xiong DY, Liu Q (2003) HHMM-based Chinese lexical analyzer ICTCLAS. *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing* (Association for Computational Linguistics, Stroudsburg, PA), Vol 17, pp 184–187.
9. McCallum A, Freitag D, Pereira FCN (2000) Maximum entropy Markov models for information extraction and segmentation. *Proceedings of the 17th International Conference on Machine Learning*, ed Langley P (Morgan Kaufmann Publishers Inc, San Francisco), Vol 17, pp 591–598.
10. Lafferty J, McCallum A, Pereira FCN (2001) Conditional random fields: Probabilistc models for segmenting and labeling sequence data. *Proceedings of the 18th International Conference on Machine Learning*, eds Brodley CE, Danyluk AP (Morgan Kaufmann Publishers Inc, San Francisco), pp 282–289.
11. Xue NW (2003) Chinese word segmentation as character tagging. *Int J Comp Linguist Chinese Lang Proc* 8(1):29–48.
12. Peng F, Feng F, McCallum A (2004) Chinese segmentation and new word detection using conditional random fields. *Proceedings of the 20th International Conference on Computational Linguistics* (Association for Computational Linguistics, Stroudsburg, PA), p 562.
13. Teahan WJ, McNab R, Wen YY, Witten IH (2000) A compression-based algorithm for Chinese word segmentation. *Comput Linguist* 26(3):375–393.
14. Ge X, Pratt W, Smyth P (1999) Discovering Chinese words from unsegmented text. *Proceeding of the 22nd ACM SIGIR Conference on Research and Development in Information Retrieval* (ACM, New York), pp 271–272.
15. Wu A, Jiang Z (2000) Statistically-enhanced new word identification in a rule-based Chinese system. *Proceedings of the Second Workshop on Chinese Language Processing: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics* (Association for Computational Linguistics, Stroudsburg, PA), Vol 12, pp 46–51.
16. Feng HD, Chen K, Deng XT, Zheng WM (2004) Accessor variety criteria for Chinese word extraction. *Comput Linguist* 30(1):75–93.
17. Li H, Huang C, Gao J, Fan X (2004) The use of SVM for Chinese new word identification. *Proceedings of the First International Joint Conference on Natural Language*, pp 497–504.
18. Wang ZR, Liu T (2005) Chinese unknown word identification based on local bigram model. *Int J Comp Proc Orient Lang* 18(3):185–196.
19. Harris ZS (1954) Distributional structure. *Word* 10:146–162.
20. Saffran JR, Newport EL, Aslin RN (1996) Word segmentation: The role of distributional cues. *J Mem Lang* 35:606–621.
21. Jelinek F (1997) *Statistical Methods for Speech Recognition* (MIT, Cambridge, MA).
22. Christiansen MH, Allen J, Seidenberg M (1998) Learning to segment speech using multiple cues: A connectionist model. *Lang Cogn Process* 13(2):221–268.
23. Olivier DC (1968) Stochastic grammars and language acquisition mechanisms. PhD thesis (Harvard University, Cambridge, MA).
24. Brent MR, Cartwright TA (1996) Distributional regularity and phonotactic constraints are useful for segmentation. *Cognition* 61(1-2):93–125.
25. Chang JS, Su KY (1997) An unsupervised iterative method for Chinese new lexicon extraction. *Int J Comp Linguist Chinese Lang Proc* 1(1):101–157.
26. Cohen P, Niall A, Brent H (2007) Voting experts: An unsupervised algorithm for segmenting sequences. *Intell Data Anal* 11(6):607–625.
27. Bussemaker HJ, Li H, Siggia ED (2000) Building a dictionary for genomes: Identification of presumptive regulatory sites by statistical analysis. *Proc Natl Acad Sci USA* 97(18):10096–10100.
28. Wang G, Yu T, Zhang W (2005) WordSpy: Identifying transcription factor binding motifs by building a dictionary and learning a grammar. *Nucleic Acids Res* 33(Web Server issue):W412–W416.
29. Agrawal R, Srikant R (1995) Mining sequential patterns. *Proceedings of the 11th International Conference on Data Engineering*, eds Yu PS, Chen ALP (IEEE Computer Society, Washington, DC), pp 3–14.
30. Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. *J R Stat Soc B* 39:1–38.
31. Akaike H (1974) A new look at the statistical model identification. *IEEE Trans Automat Contr* 19(6):716–723.
32. Schwarz G (1978) Estimating the dimension of a model. *Ann Stat* 6(2):461–464.
33. Tibshirani R (1996) Regression shrinkage and selection via the Lasso. *J R Stat Soc B* 58(1):267–288.
34. Blei D, Ng A, Jordan M (2003) Latent Dirichlet allocation. *J Mach Learn Res* 3:993–1022.
35. Griffiths TL, Steyvers M (2004) Finding scientific topics. *Proc Natl Acad Sci USA* 101 (Suppl 1):5228–5235.
36. Blei D, Lafferty J (2007) A correlated topic model of science. *Ann Appl Stat* 1(1):17–35.
37. Agrawal R, Srikant R (1994) Fast algorithms for mining association rules. *Proceedings of the 20th International Conference on Very Large Data Bases*, eds Bocca JB, Jarke M, Zaniolo C (Morgan Kaufmann Publishers Inc, San Francisco), pp 487–499.
38. Zaki MJ (2000) Generating non-redundant association rules. *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, New York), pp 34–43.
39. Han J, Pei J, Yin Y, Mao R (2004) Mining frequent patterns without candidate generation. *Data Min Knowl Discov* 8:53–87.
40. Webb G (2007) Discovering significant patterns. *Mach Learn* 68(1):1–33.
41. Deng K, Geng Z, Liu JS (2014) Association pattern discovery via theme dictionary models. *J R Stat Soc B* 76:319–347.
42. Bengio Y, Ducharme R, Vincent P, Jauvin C (2003) A neural probabilistic language model. *J Mach Learn Res* 3:1137–1155.
43. Levy O, Goldberg Y (2014) Neural word embedding as implicit matrix factorization. *Advances in Neural Information Processing Systems 2014*. Available at papers.nips.cc/paper/5477-scalable-non-linear-learning-with-adaptive-polynomial-expansions. Accessed May 4, 2016.
44. Borg I, Groenen PJ (2005) *Modern Multidimensional Scaling: Theory and Applications* (Springer Science and Business Media, New York).